Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 1 (2012) 2449–2455

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

Credit risk evaluation by using nearest subspace method

Xiaofei Zhou[a,*], Wenhan Jiang[b,c], Yong Shi[a,d]

[a]*Graduate University of Chinese Academy of Sciences, Beijing 100190, China*
[b]*First Research Institute of Ministry of Public Security, Beijing 100048, China*
[c]*Tsinghua University, Department of Electronic Engineering, Beijing 100084, China*
[d]*College of Information Science and Technology University of Nebraska at Omaha, Omaha, NE 68182, USA*
zhouxf@gucas.ac.cn, wenhan@mail.tshinghua.edu.cn, yshi@gucas.ac.cn, yshi@unomaha.edu

Abstract

In this paper, a classification method named nearest subspace method is applied for credit risk evaluation. Virtually credit risk evaluation is a very typical classification problem to identify "good" and "bad" creditors. Currently some machine learning technologies, such as support vector machine (SVM), have been discussed widely in credit risk evaluation. But there are many effective classification methods in pattern recognition and artificial intelligence have not been tested for credit evaluation. This paper presents to use nearest subspace classification method, a successful face recognition method, for credit evaluation. The nearest subspace credit evaluation method use the subspaces spanned by the creditors in same class to extend the training set, and the Euclidean distance from a test creditor to the subspace is taken as the similarity measure for classification, then the test creditor belongs to the class of nearest subspace. Experiments on real world credit dataset show that the nearest subspace credit risk evaluation method is a competitive method.

Keywords: Credit risk, credit evaluation, classification, subspace

1. Introduction

Credit risk evaluation analysis is a hot topic in the financial risk management. In fact, it is a typical classification problem to discriminate "good" and "bad" creditors. Currently many data mining techniques have been used to evaluate credit risk, such as logit analysis [1], probit analysis [2], artificial neural networks (ANN)[3][4], genetic algorithm (GA)[5], multiple criteria linear programming (MCLP)[6][7] and support vector machine (SVM)[8-14] etc. Although so many learning methods emerging in credit evaluation, there are still some effective classification methods in pattern recognition and artificial intelligence have not been tested for credit evaluation.

In this paper, we applied a nearest subspace classification idea [15][16] [17] for credit risk evaluation. The idea of nearest subspace classification uses the subspaces spanned by each class training samples to represent training set, and a query samples is classified to the class of nearest subspace. This classification idea has been successfully used in face recognition problems [16] [17]. In [16], nearest feature subspace (NFS) method is based on feature extraction which is necessary for face image data, and adopt arbitrary k (k>3) feature training samples to span subspaces in

* Corresponding author. Tel.: 0086-10-82680697; fax: 0086-10-82680698.
E-mail address: zhouxf@gucas.ac.cn.

1877-0509
doi:10.1016/j.procs.2010.04.276

each class, then these subspaces are as the extensions for training set. In [17], the subspaces spanned by all the samples per class are testified best for classification, so we also use such subspaces to represent each class. For credit evaluation, we find that current credit data are usually low dimension data, so the feature reduction procession is not adopted in this paper. That is, we use all the samples of each class to create respective subspaces, and a test sample will belong to the class represented by the nearest subspace. We call this credit evaluation method as nearest subspace (NS) credit evaluation method. On an U.S. credit dataset, compared with SVM and 1-NN method, the NS credit evaluation method is more effective and competitive.

The remainder of this paper is organized as follows: Section 2 introduces the nearest subspace algorithm. Section 3 gives the experiments on credit evaluation dataset. Finally, section 4 is the conclusion.

2. Nearest Subspace Algorithm

The basic idea of NS method is to expand representational capacity of prototypes of each class by subspace. This virtually provides an infinite number of prototypical points, and thus can account for more prototypical changes than the original samples. In the calculation of distance between a query vector and a class, the query is projected to the subspace spanned by the samples of this class. The projection point is the linear combination that is nearest to the query. The distance between the query and projection point is taken as the measure for classification.

For credit evaluation application, nearest subspace credit evaluation method uses linear combination (subspace) of all creditors belonging to the same class to approximate the possible variants of creditors. Thus the training creditor set is expanded from finite creditors to infinite linear combination of these creditors. For a test creditor, we will find a proximate credit record in each class creditor set. The proximate credit record may be not from a real creditor, but a best linear combination of all creditors in the same class. Finally, among these best creditor records from all the classes, the nearest one to the test creditor is taken as the same class creditor with the test creditor. The set of linear combination of the creditors is just the subspace spanned by creditors. Thus, NS method virtually divides a creditor to the nearest subspace of creditors. Compared with conventional 1-NN and k-NN classifiers, NS provides more possible creditors derived from original creditors. The capacity of the known creditor set is thus expanded. In the following, we describe the measure of a test creditor to a subspace, and give the NS method for credit evaluation.

2.1. Subspace Distance

In NS method, the subspace spanned by training samples of a class is taken as distribution estimation of the class, and Euclidean distance from a test sample to the subspace is taken as the similarity measure for classification. Given a set from a class $\boldsymbol{S}\subset\boldsymbol{R}^d$, $\boldsymbol{S}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_k\}$, subspace of $\boldsymbol{S}$ is the set including all linear combinations of samples in $\boldsymbol{S}$:

$$F(\boldsymbol{S})=\sum_{i=1}^{k}\alpha_i\boldsymbol{x}_i \ . \tag{1}$$

The distance function between a query $\boldsymbol{x}$ and the subspace of $\boldsymbol{S}$ can be written in detail:

$$\begin{aligned} d^2(\boldsymbol{x},F(\boldsymbol{S})) &= \min_{y\in F(\boldsymbol{S})}\left\|\boldsymbol{x}-\boldsymbol{y}\right\|^2 \\ &= \min_{\boldsymbol{\alpha}}\left\|\boldsymbol{x}-\sum_{i=1}^{k}\alpha_i\boldsymbol{x}_i\right\|^2 \\ &= \min_{\boldsymbol{\alpha}}[(\boldsymbol{x}\cdot\boldsymbol{x})-2\sum_{i=1}^{k}\alpha_i(\boldsymbol{x}\cdot\boldsymbol{x}_i)+\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)] \end{aligned} \tag{2}$$

Let $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k)$, $\boldsymbol{y}=\boldsymbol{X}\boldsymbol{\alpha}$. The Eq. (2) can be written in matrix:

$$d^2(\boldsymbol{x},F(\boldsymbol{S}))=\min_{\boldsymbol{\alpha}}\left(\boldsymbol{x}^T\boldsymbol{x}-2\boldsymbol{x}^T\boldsymbol{X}\boldsymbol{\alpha}+\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}\right) \quad (3)$$

Eq. (3) is an unconstrained optimal problem, which can be computed directly:

$$\boldsymbol{\alpha}=(\boldsymbol{X}^T\boldsymbol{X})^+\boldsymbol{X}^T\boldsymbol{x} \quad (4)$$

or

$$\boldsymbol{\alpha}=(\boldsymbol{X}^T\boldsymbol{X}+\sigma\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{x} \quad (5)$$

where $(\boldsymbol{X}^T\boldsymbol{X})^+$ is the pseudo-inverse of $\boldsymbol{X}^T\boldsymbol{X}$; $\sigma\geq 0$, and $\boldsymbol{I}$ is $k\times k$ identity Matrix.

After we compute $\boldsymbol{\alpha}$, the projection $\boldsymbol{y}$, the best linear combination of samples in this subspace $F(\boldsymbol{S})$ can be written by the coefficient $\boldsymbol{\alpha}$:

$$\boldsymbol{y}=\sum_{i=1}^{k}\alpha_i\boldsymbol{x}_i \quad (6)$$

$\boldsymbol{y}$ is the nearest point to $\boldsymbol{x}$ in subspace $F(\boldsymbol{S})$, and we take the Euclidean distance $\|\boldsymbol{x}-\boldsymbol{y}\|$ as the distance of $\boldsymbol{x}$ to $F(\boldsymbol{S})$. Then we can compute $d^2(\boldsymbol{x},F(\boldsymbol{S}))$ by $\boldsymbol{y}$:

$$d^2(\boldsymbol{x},F(\boldsymbol{S}))=\|\boldsymbol{x}-\boldsymbol{y}\|^2 \quad (7)$$

2.2. Nearest Subspace Algorithm

Given a multi-class training set $\boldsymbol{S}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\ldots,\boldsymbol{S}_l\}$, $\boldsymbol{S}_i\subset\mathbf{R}^d$ is the ith class training set. The l subspaces from different category training sets are $F(\boldsymbol{S}_1)$,…, $F(\boldsymbol{S}_l)$. For an arbitrary query $\boldsymbol{x}\subset\mathbf{R}^d$, we separately compute the distance between $\boldsymbol{x}$ and each class subspace:

$$d^2(\boldsymbol{x},F(\boldsymbol{S}_1))=\min_{\boldsymbol{y}_1\in F(\boldsymbol{S}_1)}\|\boldsymbol{x}-\boldsymbol{y}_1\|^2,$$

…,

$$d^2(\boldsymbol{x},F(\boldsymbol{S}_l))=\min_{\boldsymbol{y}_l\in F(\boldsymbol{S}_l)}\|\boldsymbol{x}-\boldsymbol{y}_l\|^2.$$

We take $d^2(\boldsymbol{x},F(\boldsymbol{S}_i))$ as the similarity of $\boldsymbol{x}$ and the ith class, and classify $\boldsymbol{x}$ to the class of the nearest neighbor subspace. That is, $\boldsymbol{x}$ belongs to the class $\boldsymbol{S}_j$, $\boldsymbol{S}_j=\arg\min_{\boldsymbol{S}_i}d^2(\boldsymbol{x},F(\boldsymbol{S}_i))$, $j=1,2,\cdots,l$.

Following, for a test creditor $\boldsymbol{x}$, we give the overall process of NS method for the creditor's evaluation:

Step 1: Computing the optimal weights $\boldsymbol{\alpha}$ for each class creditors.

For the ith class creditor set $\boldsymbol{S}_i=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k\}$, the weights $\boldsymbol{\alpha}$ for the best linear combination of $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k$ can be computed directly:

$$\boldsymbol{\alpha}=(\boldsymbol{X}^T\boldsymbol{X})^+\boldsymbol{X}^T\boldsymbol{x} \text{ or } \boldsymbol{\alpha}=(\boldsymbol{X}^T\boldsymbol{X}+\sigma\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{x}$$

where $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k)$, $(\boldsymbol{X}^T\boldsymbol{X})^+$ is the pseudo-inverse of $\boldsymbol{X}^T\boldsymbol{X}$; $\sigma\geq 0$, and $\boldsymbol{I}$ is $k\times k$ identity Matrix.

Step 2: Finding the best linear combination record of creditors in each class.

For the ith class creditor set $\boldsymbol{S}_i=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k\}$, by the weights $\boldsymbol{\alpha}$, we can obtain the projection $\boldsymbol{y}_i$ of the test

creditor $\boldsymbol{x}$ in subspace $F(\boldsymbol{S}_i)$:

$$\boldsymbol{y}_i=\sum_{i=1}^{k}\alpha_i\boldsymbol{x}_i \ .$$

Step 3: Computing the distance of the test creditor $\boldsymbol{x}$ to each subspace.

The distance between $\boldsymbol{x}$ and ith class subspace $F(\boldsymbol{S}_i)$ is

$$d^2(\boldsymbol{x},F(\boldsymbol{S}_i))=\left\|\boldsymbol{x}-\boldsymbol{y}_i\right\|^2$$

Step 4: Classifying the test creditor $\boldsymbol{x}$ to the nearest subspace

We find the minimal distance among all the distances $d^2(\boldsymbol{x},F(\boldsymbol{S}_1)),\ldots,d^2(\boldsymbol{x},F(\boldsymbol{S}_l))$, and decide the test creditor $\boldsymbol{x}$ to the class of the nearest subspace: $\boldsymbol{x}$ belongs to the class of $\boldsymbol{S}_j$, $\boldsymbol{S}_j=\arg\min_{\boldsymbol{S}_i} d^2(\boldsymbol{x},F(\boldsymbol{S}_i))$, $j=1,2,\cdots,l$.

3. Credit Evaluation Experiments

Credit risk evaluation is a very typical classification problem to identify "good" and "bad" creditors[18][19]. In this paper, we apply nearest subspace method for credit risk evaluation. To test the efficacy of NS for creditor evaluation, we compare it with SVM by linear kernel and RBF kernel (k=exp (-0.5($\| x\text{-}y \| / \sigma$)2)) on a real world US credit dataset. The credit card dataset used in our experiments is provided by a major U.S. bank. It contains 6000 records and 66 derived attributes. Among these 6000 records, 960 are bankruptcy accounts and 5040 are "good" status accounts [18]. In our experiments, three accuracies will be tested to evaluate the classifiers, "Good" accuracy, "Bad" accuracy and Total accuracy:

$$\text{"Good" Accuracy}=\frac{\text{number of correctly classified "Good" samples in test set}}{\text{number of "Good" samples in test set}},$$

$$\text{"Bad" Accuracy}=\frac{\text{number of correctly classified "Bad" samples in test set}}{\text{number of "Bad" samples in test set}},$$

$$\text{Total Accuracy}=\frac{\text{number of correct classification in test set}}{\text{number of samples in test set}}.$$

where "Good" accuracy and "Bad" accuracy respectively measure the capacity of the classifiers to identify "Good" or "Bad" clients. In the real world, for the special purposes to prevent the credit fraud, the accuracy of classification for the risky class must be improved to reach an acceptable standard but not excessively affecting the accuracy of classification for other classes. Thus, improving "Bad" accuracy is one of the most important tasks in credit scoring.

In our experiments of each dataet, we random select p (p=10, 20, 30,…, 100) samples from each class as training set and the remaining for test. We repeat the each classifiers test 20 times and report the mean results. All of our experiments are carried out on Matlab 7.0 platform. The convex quadratic programming problem of SVM is solved utilizing Matlab optimal tools. The "Bad", "Good" and total accuracy comparisons of the classifiers are shown in Table 1, Table 2 and Table 3 respectively. Parameter σ of RBF kernel of SVM and KASNP is σ=10000, and the penalty constant C of SVM is C=∞.

Table 1 “Bad” accuracy (%) comparisons of different methods on US dataset

Number of training data per class	“Bad” accuracy (%) comparisons on USA dataset			
	1-NN	Linear SVM	RBF SVM	NS
10	66.67 %	59.68%	64.51%	58.75 %
20	63.37 %	66.23%	65.43%	65.03 %
30	64.38 %	65.25%	64.48%	72.13 %
40	63.77 %	63.97%	65.34%	76.83 %
50	65.78 %	65.21%	66.20%	76.32 %
60	64.82 %	65.82%	66.01%	74.32 %
70	65.52 %	65.89%	68.31%	***75.22 %***
80	65.46 %	67.37%	69.99%	***72.14 %***
90	65.60 %	66.94%	70.62%	***71.99 %***
100	64.83 %	66.32%	70.69%	***71.62 %***

Table 2 “Good” accuracy (%) comparisons of different methods on US dataset

Number of training data per class	“good” accuracy (%) comparisons on USA dataset			
	1-NN	Linear SVM	RBF SVM	NS
10	56.48 %	60.97%	61.50%	***66.36 %***
20	59.40 %	66.60%	66.82%	***69.89 %***
30	59.83 %	65.03%	65.64%	***67.23 %***
40	62.41 %	67.12%	***67.62%***	64.43 %
50	61.55 %	66.46%	***66.62%***	65.81 %
60	62.45 %	66.46%	***67.84%***	67.33 %
70	62.65 %	66.65%	67.49%	***68.73 %***
80	62.15 %	66.65%	66.35%	***71.24 %***
90	62.54 %	66.67%	67.74%	***69.70 %***
100	63.23 %	67.02%	67.97%	***68.44 %***

Table 3 Total accuracy (%) comparisons of different methods on US dataset

Number of training data per class	Total accuracy (%) comparisons on USA dataset			
	1-NN	Linear SVM	RBF SVM	NS
10	58.10%	60.77%	61.97%	***65.15%***
20	60.03%	66.55%	66.60%	***69.13%***
30	60.54%	63.83%	65.46%	***67.99%***
40	62.62%	***67.81%***	67.27%	66.36%
50	62.20%	66.27%	66.55%	***67.43%***
60	62.81%	66.44%	67.56%	***68.40%***
70	63.08%	66.54%	67.61%	***69.72%***
80	62.65%	67.39%	66.90%	***71.38%***
90	62.99%	65.13%	68.17%	***70.04%***
100	63.46%	66.92%	68.37%	***68.91%***

In our experiments, NS method made a better risky classification performance. Comparing the results reported in Table 1 to Table 3, we find the NS method can keep three accuracies “bad”, “good” and “Total” at better standard. (1) For finding “Bad” clients (nearest subspace (NS) method is superior to other classifiers. As we can see from Table 1, when the training samples per class is greater than 30, NS method all can achieve above 70% classification accuracy, whereas the accuracies of other methods are mostly less than 70%. (2) For identifying “Good” clients (see

Table 2), NS can obtain highest accuracy in seven compared experiments when p=10,20,30,70,80,90,100, and RBF SVM have three highest accuracies when p=40,50,60. (3) From the general view (see Table 3), NS method dominates 1-NN and SVMs.

Thus, from above experimental results of the U.S credit dataset, we conclude that the NS method is comparable with 1-NN and SVMs for creditor classification.

4. Conclusion

This paper introduces a classification method, nearest subspace method, for credit evaluation. The nearest subspace credit evaluation method uses linear combination (subspace) of all creditors belonging to the same class to approximate the possible variants of creditors. For a test creditor, NS method is to find a best approximation from the nearest subspace, and divides the test creditor to the class of nearest subspace. On a real world U.S. credit card dataset, the NS shows good performance for credit evaluation.

Acknowledgements

This work was partially supported by the Post Doctor National Grand Fundamental Project of China (No.20090450607), the National Grand Fundamental Research 973 Program of China under Grant No.2004CB720103, by the National Nature Science Foundation of China under Grant No.70531040, No. 70921061, No.10601064 and No.70501030 and by a research grant from BHP Billion Co., Australia.

References

1. S. Scholes, Discuss. Faraday Soc. No. 50 (1970) 222.

2. O.V. Mazurin and E.A. Porai-Koshits (eds.), Phase Separation in Glass, North-Holland, Amsterdam, 1984.

3. Y. Dimitriev and E. Kashchieva, J.Mater. Sci. 10 (1975) 1419.

4. D.L. Eaton, Porous Glass Support Material, US Patent No. 3 904 422 (1975).

1. Wiginton, J. C. A note on the comparison of logit and discriminant models of consumer credit behaviour. Journal of Financial Quantitative Analysis, 15 (1980), 757-770.

2. Grablowsky, B. J., & Talley, W. K. Probit and discriminant functions for classifying credit applicants: A comparison. Journal of Economic Business, 33 (1981), 254-261.

3. Malhotra, R., & Malhotra, D. K. Evaluating consumer loans using neural networks. Omega, 31, 83-96.

4. Smalz, R., & Conrad, M. (1994). Combining evolution with credit apportionment: A new learning algorithm for neural nets. Neural Networks, 7 (2003), 341 – 351.

5. Varetto, F. Genetic algorithms applications in the analysis of insolvency risk. Journal of Banking and Finance, 22 (1998), 1421-1439.

6. Shi, Y., Peng, Y., Xu, W., &Tang, X. Data Mining via Multiple Criteria Linear Programming: Applications in Credit Card Portfolio Management, International Journal of Information Technology and Decision Making, Vol. 1 (2002), 131-151.

7. Shi, Y., Wise, M., Luo, M., & Yu, L. Data mining in credit card portfolio management: a multiple criteria decision making approach, in Koksalan, M. and Zionts, S. eds., Multiple Criteria Decision Making in the New Millennium, Springer, Berlin, (2001), 427-436.

8. Vapnik, V. N. The Nature of Statistical Learning Theory. New York: Springer-Verlag, (1995).

9. Van Gestel, T., Baesens, B., Garcia, J., & Van Dijcke, P.: A support vector approach to credit scoring. Bank en Financiewezen 2 (2003): 73-82.

10. Bellotti, T., & Crook, J. Support vector machines for credit scoring and discovery of significant features. Expert Systems with Applications, 36(2), (2009) 3302 – 3308.

11. Lai, K. K., Yu, L., Zhou, L. G., & Wang, S. Y. Credit risk evaluation with least square support vector machine. Lecture Notes in Artificial Intelligence, 4062 (2006), 490-495.

12. Yu, L., Wang, S. Y., Lai, K. K., & Zhou, L. G. Bio-inspired credit risk analysis computational intelligence with support vector machines. Berlin: Springer-Verlag (2008).

13. Yu, L., Wang, S., Cao, J. A modified least squares support vector machine classifier with application to credit risk analysis.International Journal of Information Technology & Decision Making, Volume: 8, Issue: 4 (2009), Page: 677-696.

14. Zhou, L. Lai, K. K. and Yen, J. Credit scoring models with auc maximization based on weighted SVM. International Journal of Information Technology & Decision Making, Volume: 8, Issue: 4 (2009), Page: 677-696.

15. Oja, E. (1983). Subspace Methods of Pattern Recognition. Letchworth, England: Research Studies Press Ltd.

16. Chien, J. T., & Wu, C. C. (2002). Discriminant Waveletfaces and Nearest Feature Classifiers for Face Recognition, IEEE Trans. On PAMI, Vol.24, No.12, 1644-1649.

17. Li, S. Z. (1998). Face recognition based on nearest linear combinations, Computer Vision and Pattern Recognition, Proceedings of 1998 IEEE Computer Society Conference on, Page(s): 839-844.

18. Shi, Y., Peng, Y. Kou, G., Chen, Z.X. Classifing Credit card accounts for business intelligence and decision making: A multiple-creteria quadratic programming approach. International Journal of Information Technology & Decision Making, Volume: 4, Issue: 4(2005) pp. 581-599.

19. Shi, Y. The research trend of information technology and decision making in 2009.International Journal of Information Technology & Decision Making,,Volume: 9, Issue: 1(2010), 1-8.

20. He, J., Liu, X. T., Shi, Y., Xu, W. X., & Yan, N. Classifications of Credit Cardholder Behavior by Using Fuzzy Linear Programming, Information Technology and Decision Making, 3(2004), pp.633-650.